An Indexing and Searching Structure for Multimedia Database Systems

Shu-Ching Chen[a], Srinivas Sista[c], Mei-Ling Shyu[b], and R. L. Kashyap[b]

[a]School of Computer Science, Florida International University, Miami, FL 33199
[b]School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907
[c]EVision LLC, Northbrook, IL 60062

ABSTRACT

Recently, multimedia database systems have emerged as a fruitful area for research due to the recent progress in high-speed communication networks, large capacity storage devices, digitized media, and data compression technologies over the last few years. Multimedia information has been used in a variety of applications including manufacturing, education, medicine, entertainment, etc. A multimedia database system integrates text, images, audio, graphics, animation, and full-motion video in the application environments. The important characteristic of a multimedia database system is that all of the different media are brought together into one single unit, all controlled by a computer. As more information sources become available in multimedia systems, how to model and search the information efficiently is very crucial. In this paper, we present a database searching structure that incorporates image processing techniques to model multimedia data. A *Simultaneous Partition and Class Parameter Estimation (SPCPE)* algorithm that considers the problem of video frame segmentation as a joint estimation of the partition and class parameter variables has been developed and implemented to identify objects and their corresponding spatial relations. Based on the obtained object information, a *web spatial model (WSM)* is constructed. A *WSM* is a multimedia database searching structure to model the temporal and spatial relations of semantic objects so that multimedia database queries related to the objects' temporal and spatial relations on the images or video frames can be answered efficiently.

Keywords: Multimedia database systems, Indexing, Database searching, Video segmentation

1. INTRODUCTION

Unlike the traditional relational database systems which consist only of alphanumeric data, the multimedia database systems not only have the alphanumeric data but also the data that covers multi-dimensional spaces such as audio, images and videos. In multimedia systems, a variety of information sources – text, voice, image, audio, animation, and video – are delivered synchronously or asynchronously via more than one device. The important characteristic of such a system is that all of the different media are brought together into one single unit, all controlled by a computer. Normally, multimedia systems require the management and delivery of extremely large bodies of data at very high rates and may require the delivery with real-time constraints. In traditional database management systems (DBMS), such as relational database systems, only text information is stored in the database and there is no need to consider the synchronicity among media. In object-oriented database systems, a database may include image data and the DBMS still is not designed to support multimedia information. Multimedia extension is needed to handle the mismatch between multimedia data and the conventional object-oriented database management systems.[3] In multimedia database systems, a new design of multimedia database management systems (MDBMS) is required to handle the temporal and spatial requirements, and the rich semantics of multimedia data such as text, image, audio, and video. The temporal requirements are that media needs to be synchronous and to be presented at the specified time that was given at authoring time. The spatial requirement is that the DBMS needs to handle the layout of the media at a certain point in time. For image and video frames, the DBMS needs to keep the relative positions of semantic objects (building, car, etc.) so that users can issue queries, such as, "Find a video clip that has one car

Further author information: (Send correspondence to Shu-Ching Chen)
Shu-Ching Chen: E-mail: chens@cs.fiu.edu
Srinivas Sista: E-mail: sista@evisionglobal.com
Mei-Ling Shyu: E-mail: shyu@ecn.purdue.edu
R. L. Kashyap: E-mail: kashyap@ecn.purdue.edu

in front of a building." However, extracting information from images/videos is time consuming. In order to provide fast response for real time applications, information or knowledge needs to be extracted from images/videos item by item in advance and stored for later retrieval. For example, to do spatial reasoning we would have to store numerous spatial relations among objects.[2] Extracting object information from images/videos can be achieved by image and video segmentation techniques.

With the emerging demand on content based video processing approaches, more and more attention is devoted to segmenting video frames into regions such that each region, or a group of regions, corresponds to an object that is meaningful to human viewers.[7,6] This kind of object based representation of the video data is being incorporated into standards like MPEG4 and MPEG7.[7] A video clip is a temporal sequence of two dimensional samples of the visual field. Each sample is an image which is referred to as a frame of the video. Segmentation of an image, in its most general sense, is to divide it into smaller parts. In image segmentation, the input image is partitioned into regions such that each region satisfies some homogeneity criterion. The regions, which are usually characterized by homogeneity criteria like intensity values, texture, etc., are also referred to as *classes*. Video segmentation is a very important step in processing video clips. One of the emerging applications in video processing is its storage and retrieval from multimedia databases and content based indexing. Video data can be temporally segmented into smaller groups depending on the scene activity where each group contains several frames. Clips are divided into scenes and scenes into shots. A shot is considered the smallest group of frames that represent a semantically consistent unit.

To date, there are very few methods of image segmentation that addressed partitioning and obtaining content description of segments simultaneously.[1,8,10] In,[1] the problem was posed as texture segmentation where the textures are modeled by Gauss-Markov random fields. Horn and Schunck proposed a smoothness constraint where the motion field varies smoothly in most part of the image.[8] In,[10] the problem was posed as segmentation of Gibbs random fields and solved using simulated annealing. However, our proposed method recognizes the variability of content description depending on the complexity of the image regions and effectively addresses it. We introduce the notion of a class as that which gives rise to different segments with the same content description. In particular, our framework allows us to partition the data as well as obtain descriptions of classes for a large family of parameter models. These parameter models are used to describe the content of the class. Central to our method is the formulation of a cost functional defined on the space of image partitions and the class description parameters that can be minimized in a simple manner.

As more information sources become available in multimedia systems, the knowledge embedded in images or videos, especially spatial knowledge, should be captured by the data structure as much as possible. For this purpose, an unsupervised video segmentation method, the *Simultaneous Partition and Class Parameter Estimation (SPCPE)* algorithm, and a multimedia database searching structure called *web spatial model (WSM)* are incorporated together in this paper. The objective of the *SPCPE* algorithm is to obtain objects in each video frame and their corresponding spatial relations[9,11]; while the objective of the *WSM* is to model the spatial relations among objects, each covered by a bounding box. In the *SPCPE* algorithm, each frame is partitioned into several object regions using the partition of the previous frame as an initial condition. So the correspondence problem need not be addressed explicitly since the information from the previous frame essentially guides the partitioning of the current frame. Our interest is in obtaining object level segmentation in the proposed *SPCPE* algorithm.

A *WSM* is a multimedia database searching structure which consists of a set of nodes and a set of links. A *WSM* organizes the spatial relations among the semantic objects (e.g., a car in an image or a video frame) into a structural construct. It helps to identify the spatial relations of the semantic objects required in a query. The basic twenty-seven spatial relations introduced in[4,5] are used in the *WSM* to model the objects' spatial relations. Based on the object information provided by the video segmentation method, the *WSM* can structure the temporal and spatial relations of semantic objects so that the multimedia database queries that involve objects' temporal and spatial relations on the images or video frames can be answered efficiently.

The organization of this paper is as follows. Section 2 introduces the video segmentation method with an example soccer game video. In section 3, the *WSM* is presented. Section 4 shows how to use the *WSM* to answer the multimedia database queries. This paper is summarized in section 5.

2. VIDEO FRAME SEGMENTATION

To partition each video frame, we employ a descent algorithm, called the *simultaneous partition and class parameter estimation (SPCPE)* algorithm, that minimizes a functional defined over the discrete space of image partitions to yield an estimate of the optimal partition as well as the class description parameters. The proposed video frame segmentation method starts with an arbitrary partition and employs the *SPCPE* algorithm iteratively to estimate the partition and the class description parameters jointly.

2.1. Classes and Segments within Each Frame

Traditionally, an image is divided into chunks of connected pixels so that two distinct chunks have distinct meaning. However, in real images, like the Landsat images or aerial views of urban areas, there could be hundreds of segments depending on how we view a segment. So a single class can contain several disconnected segments. Given an image, our aim is to discover the different categories in it and obtain the various segments that belong to each one of these categories. So, we view the problem as a partitioning/segmentation problem.

Here, we first clarify the concepts of a class and a segment.

- A *class* is characterized by a statistical description and consists of all the regions in an image that follow this description. For example, houses, roads, parks, etc. form classes.
- A *segment* is an instance of a class. For example, the actual occurrence of a class in the image are the various segments.

2.2. Simultaneous Partition and Class Parameter Estimation (SPCPE) Algorithm

The definitions of classes and segments that we introduced in the context of partitioning image data carry over directly with some modifications to account for the temporal dimension processed by video data. In a video, the successive frames do not differ much due to the high temporal sampling rate. Hence, the partions of adjacent frames do not differ significantly. So starting with the estimated partition of the previous frame, we may obtain a new partition that is not significantly different from the partition of the previous frame. The key idea is then to use the unsupervised image segmentation method successively on each frame of the video, incorporating the partition information of the previous frame as initial condition while partitioning the current frame. The partition and the associated class parameters are intimately related. A given class description determines a partition given by the classification scheme chosen by the user. Similarly, a given partition gives rise to a class description computed by the parameterization and the estimation method chosen by the user. Hence, the problem of video frame segmentation is posed as a joint estimation of the partition and class parameter variables.

2.2.1. Parametrizing the Classes

Suppose we know the class identities of the pixels. Then we choose some functional description for the family of classes and estimate the associated parameters of each class from the image data. As before, let us assume that that the image is of size $N_r \times N_c$ with intensities given by $Y = \{y_{ij} : 1 \leq i \leq N_r, 1 \leq j \leq N_c\}$ and that there are two classes in the image. We first make a few observations regarding the parametrization of the classes. All we have is pixel data from the image and the class identities. The mathematical description of a class specifies the pixel values y_{ij} as functions of the spatial coordinates of the pixel or as functions of its neighboring pixel values. The number of pixels in each class is large, usually much larger than the number of parameters that need to be estimated. So we have an overdetermined system of equations. If the equations are such that they form a linearly parametrizable system, the parameters of each class can be computed directly using a least squares technique.

For example, suppose we use a family of 2D polynomial functions to describe the classes. Let the pixels in class k be described by a function of the type

$$y_{ij} \quad = \quad a_k^T v_{ij}, \quad \forall (i,j)\ y_{ij} \in \boldsymbol{c}_k, \quad k = 1,2 \tag{1}$$

where a_k are the parameters of the class k and v_{ij} is the vector whose components are functions of spatial coordinates. Since we assumed that we know which class each pixel belongs to, we can easily estimate the parameters of each class using the least squares technique.

In our method, we assume that the pixels are clustered around 2D polynomials. We also assume that the errors in modeling y_{ij} in class k are zero mean i.i.d Gaussian random variables with variance ρ_k. The parameters a_k are assumed to be independent random variables with uniform prior densities. Under these assumptions, we can show that the MAP estimate of a_k will be the same as the least squares estimate.

So we have a way of parametrizing the classes and estimating their parameters if we know an a priori partition of the image. Similarly we have a way of estimating a partition if we have the class descriptions specified. However, both the partition and the class description parameters are unknown in most of the problems. In that case we have to estimate them jointly or simultaneously.

2.2.2. Joint Estimation

Denote the space of distinct partitions by Ω. Let the number of classes be 2. Each element of Ω represents a distinct partition of the image into two classes. The size of the space Ω grows exponentially with the number of pixels. Let the partition variable be $\boldsymbol{c} = \{\boldsymbol{c}_1, \boldsymbol{c}_2\}$ that takes values in the space of partitions Ω. Hence a particular value of $\boldsymbol{c}$ corresponds to a distinct partition in Ω. In the context of Bayesian estimation, $\boldsymbol{c}$ is treated as a random variable that needs to be estimated. Suppose we use a family of 2D polynomial functions parametrized by a_k to describe the class k, where the error has a Gaussian distribution with mean zero and variance ρ_k for class k. The parameter estimates of class k, $\hat{a}_k$, can be computed directly using least squares estimation.

Now, the estimates of $\boldsymbol{c} = \{\boldsymbol{c}_1, \boldsymbol{c}_2\}$ and $\boldsymbol{\theta} = \{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2\}$ are given by

$$\begin{aligned} (\hat{\boldsymbol{c}}, \hat{\boldsymbol{\theta}}) &= \operatorname*{Arg\,max}_{(\boldsymbol{c},\boldsymbol{\theta})} P(\boldsymbol{c}, \boldsymbol{\theta} \mid Y) \\ &= \operatorname*{Arg\,max}_{(\boldsymbol{c},\boldsymbol{\theta})} P(Y \mid \boldsymbol{c}, \boldsymbol{\theta}) P(\boldsymbol{c}, \boldsymbol{\theta}). \end{aligned} \tag{2}$$

We need to characterize the various probability densities in the above expression. The following assumptions are made about the partition variable $\boldsymbol{c}$, the parameters $\boldsymbol{\theta}$ and their prior distributions to simplify the problem:

1. $\boldsymbol{c}$ and $\boldsymbol{\theta}$ are independent.
2. All the partitions in Ω have equal probability; i.e., $P(\boldsymbol{c}) = 1/\#\Omega$
3. All the parameters are independent; i.e.,

$$p(\boldsymbol{\theta}) = p(\boldsymbol{\theta}_1) p(\boldsymbol{\theta}_2) = \left(\prod_{j=0}^{3} p(a_{1j}) \right) \left(\prod_{0}^{3} p(a_{2j}) \right) \tag{3}$$

   and each parameter has a uniform distribution.

Under these assumptions, the expression in (2) becomes

$$(\hat{\boldsymbol{c}}, \hat{\boldsymbol{\theta}}) = \operatorname*{Arg\,max}_{(\boldsymbol{c},\boldsymbol{\theta})} P(Y \mid \boldsymbol{c}, \boldsymbol{\theta}). \tag{4}$$

Next we assume that the pixels are statistically independent. The joint probability of the pixel data Y will then be a product of the marginal densities. We take the negative logarithm of the argument to be maximized and convert the maximization of a product of terms to the minimization of a sum of terms. Let $J(\boldsymbol{c}, \boldsymbol{\theta})$ denote the functional that needs to be minimized, i.e., the sum of terms. The expression for the estimate is given by

$$\begin{aligned} (\hat{\boldsymbol{c}}, \hat{\boldsymbol{\theta}}) &= \operatorname*{Arg\,min}_{(\boldsymbol{c},\boldsymbol{\theta})} J(\boldsymbol{c}_1, \boldsymbol{c}_2, \boldsymbol{\theta}_1, \boldsymbol{\theta}_2) \\ J(\boldsymbol{c}_1, \boldsymbol{c}_2, \boldsymbol{\theta}_1, \boldsymbol{\theta}_2) &= \sum_{y_{ij} \in \boldsymbol{c}_1} -\ln p_1(y_{ij}; \boldsymbol{\theta}_1) + \sum_{y_{ij} \in \boldsymbol{c}_2} -\ln p_2(y_{ij}; \boldsymbol{\theta}_2). \end{aligned} \tag{5}$$

Note that, in order to minimize J, we just have to assign each pixel y_{ij} to a class which yields the least value of $-\ln p_k(y_{ij}),\ k = 1, 2$. Hence, we use the following decision rule to assign the pixels y_{ij} to the classes $\boldsymbol{c}_1$ and $\boldsymbol{c}_2$

$$\begin{aligned} y_{ij} &\in \hat{\boldsymbol{c}}_1 \text{ if } -\ln p_1(y_{ij}) \leq -\ln p_2(y_{ij}) \\ &\in \hat{\boldsymbol{c}}_2 \text{ otherwise} \end{aligned} \tag{6}$$

This assignment leads to the unique global minimum of J yielding the MAP estimate of the partition variable $\boldsymbol{c}$. Note that even though the size of Ω is very large, we have no problem in estimating the partition variable. We now present the descent algorithm in its entirity.

Descent Algorithm

Let $\boldsymbol{c}^{(j)} = (\boldsymbol{c}_1^{(j)}, \boldsymbol{c}_2^{(j)})$ and $\boldsymbol{\theta}^{(j)} = (\boldsymbol{\theta}_1^{(j)}, \boldsymbol{\theta}_2^{(j)})$ be the estimates of the partition and the corresponding parameters at the end of j^{th} iteration.

1. Choose the starting segmentation $\boldsymbol{c}^{(0)}$ arbitrarily, perhaps from a solution of a clustering algorithm with random seeds.
2. (Step 1) Given $\boldsymbol{c}^{(j)}$, compute $\boldsymbol{\theta}^{(j)}$ using the method of least squares.
3. (Step 2) Given $\boldsymbol{\theta}^{(j)}$, compute $\boldsymbol{c}^{(j+1)}$ using the decision rule in (6).
4. Stop if $\boldsymbol{c}^{(j)} = \boldsymbol{c}^{(j+1)}$; otherwise goto 2.

End.

2.2.3. Partitioning each frame

The *SPCPE* algorithm starts with an arbitrary partition and computes the corresponding class parameters. From these class parameters and the data, a new partition is estimated. Both the partition and the class parameters are iteratively refined until there is no further change in them. So the minimum we obtain through our descent method depends strongly on the starting point or the initial partition. In video data, in the absence of scene changes, consecutive frames do not differ much in content. Consequently, the partitions of adjacent frames are close to each other. So, in our method, each frame is partitioned using the partition of the previous frame as an initial condition. An added advantage of this approach is that the correspondence problem need not be addressed explicitly.

For the first frame, since there is no previous frame, we use a randomly generated initial partition. Alternately, a partition generated from another clustering algorithm can be employed. Specifically, let the current frame be k. Let the estimated partition of the $(k-1)^{th}$ frame be $\boldsymbol{c}^*(k-1)$. Then we set the initial partition of the k^{th} frame, denoted by $\boldsymbol{c}^{(0)}(k)$, equal to the estimated partition of the $(k-1)^{th}$ frame

$$\boldsymbol{c}^{(0)}(k) = \boldsymbol{c}^*(k-1). \tag{7}$$

This choice not only reduces the number of iterations needed to converge to the minimum but also helps to converge to a partition that is close to that of the previous frame.

The video segmentation method is applied to an example soccer video. From the results on frames 1 through 60, a few frames – 1, 6 and 12 – are shown in Figure 1 along with the original frames adjacent to them. The centroid of each segment is marked with an 'x' and the segment is shown with a bounding box around it.

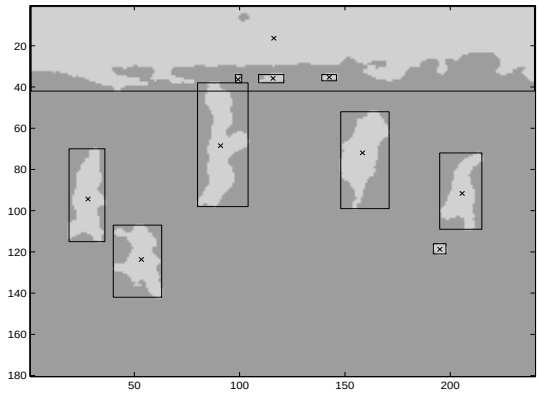
3. WEB SPATIAL MODEL (WSM)

The web spatial model (WSM) plays an important role in the database queries that involve the spatial relations. This structure can help to identify which semantic objects have the spatial properties required by the queries. WSM is a multimedia database searching structure (N, L), where N is a set of nodes and L is a set of links.

3.1. Properties of the WSM

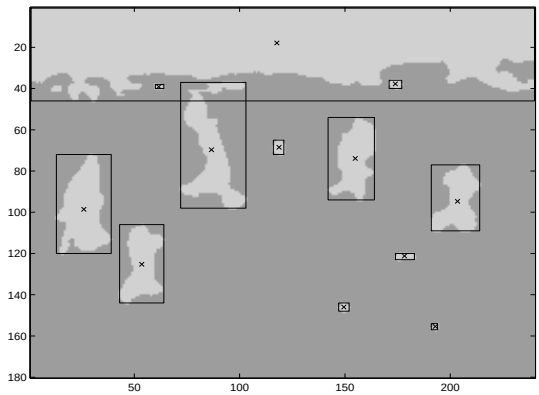
There are two important properties of the Web structure. First, it allows non-unique parent nodes. Second, it allows parallel searches and concurrent browsing paths. The details are discussed in the following two subsections.

(a) Frame 1



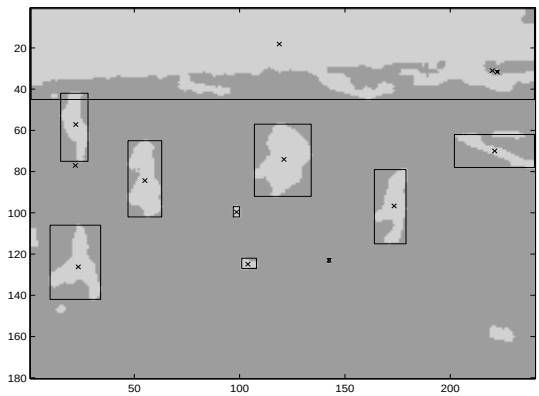
(b) Partition of Frame 1

(c) Frame 6



(d) Partition of Frame 6

(e) Frame 12



(f) Partition of Frame 12

Figure 1. Figures (a),(c),(e) are the original Frames 1,6,12 (on the left) and (b),(d),(f) show their corresponding segments (on the right). The centroid of each segment is marked with an 'x' and the segment is shown with a bounding box around it.

3.1.1. Non-unique Parents

In a web, some nodes may have more than one parent as shown in Figure 2. In Figure 2, there are five root nodes representing the temporal nodes. The nodes under the root nodes excluding the bottom nodes are the intermediate nodes. The bottom nodes represent the media nodes. Each intermediate node has only one parent node while each media node can have more than one parent node. There are two advantages to this property. The first advantage is

that the number of media nodes can be reduced. The second one is that it allows parallel searches and concurrent browsing paths discussed in the following subsection.

3.1.2. Allows Parallel Searches and Concurrent Browsing Paths

In Figure 2, the Web structure allows parallel searches a nd browsings beginning from seven root nodes. This is the main difference between a traditional tree structure and a Web structure. Unlike tree structures in which the search path is unique every time – and if we cannot find an object in a specific path then we need to go back and try another path – the web structure allows multiple search paths concurrently. In this case, the search time is reduced which is a big advantage in database queries.

3.2. Node Types

There are three types of nodes – a *spatial node*, *intermediate node*, and *semantic object node* – which are connected by the *connection link* and the *ordered link*. Each type of node forms a layer in a WSM (as shown in Figure 2). The link is used to represent the connections and the relations between the nodes, and the object information structured in a WSM is provided by the video segmentation method introduced in the previous section. The types of nodes and links are defined as follows.

1. *spatial node*: The twenty-seven spatial relations are represented by each spatial node. These nodes are the root nodes in the web spatial structure. They have no incoming link and can have more than one outgoing link to their children nodes. For example, the root node with number 1 or 10, which represents the centroid of the semantic object, is in the same region as that of the target semantic object or on the left of the semantic object, respectively.

2. *connection link*: The *connection link* connects a *spatial* node and an *intermediate* node.

3. *intermediate node*: These nodes are used to connect the *spatial* node and the semantic object node. Each *intermediate* node has only one incoming link from the root node and has two outgoing links to connect the *semantic object* nodes.

The information stored in each node is defined in the following definition:

Definition 1: Let O be a set of n semantic objects such that O = $(o_1, o_2, \ldots, o_n)$. Each intermediate node is associated with a pair that consists of two semantic object nodes o_i, o_j $\forall i, j (1 \leq i \leq n, 1 \leq j \leq n, i \neq j)$. The spatial relation S to this pair is o_i S o_j. R = $\{(m_1,(sf_1,ef_1)), (m_1,(sf_1,ef_1)), \ldots\}$ is a set of pairs for each intermediate node. Associated with each $(m_k,(sf_k,ef_k))$, $\forall k, (1 \leq k \leq n)$, is a single image frame for an image media stream m_k or a range of video frames for video stream m_k that goes from frame number sf_k to ef_k. For the image media stream, $sf_k = ef_k$.

3.3. Node Types

There are three types of nodes in WTM: the *temporal node*, the *intermediate node* and the *media stream node*.

1. *ordered link*: The *ordered link* connects an *intermediate* node and a *semantic object* node. The links are numbered by **1** and **2**. The links with number **1** and number **2** point to the *semantic object* and the *target semantic object*, respectively.

2. *semantic object node*: These nodes represent the semantic objects. They are the leaf nodes of the web spatial structure.

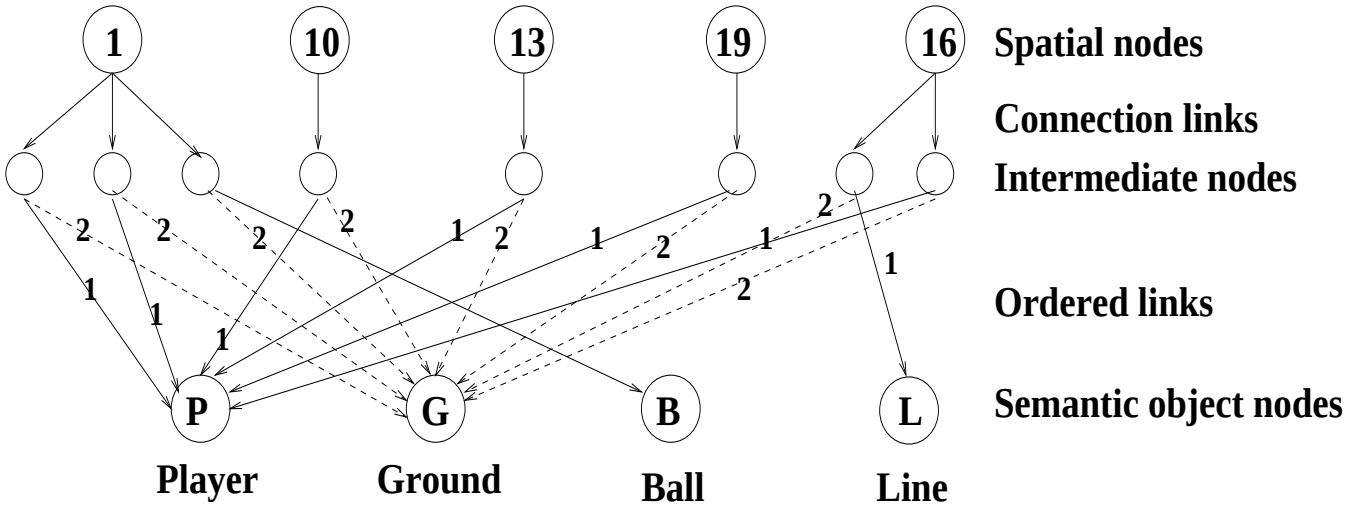


Figure 2. Web spatial relation model for semantic objects. Ordered links with number 1 (arrows) and number 2 (dashed arrows) point to the semantic objects and target semantic object node **(Ground)**.

4. MULTIMEDIA SPATIAL DATABASE QUERIES USING WSM

WSM can help to answer spatial multimedia database queries. Figure 2 is a WSM to model spatial relations in Figure 1. For simplicity, Figure 2 shows only the case when the **Ground** is selected as the target semantic object. Also, the segment for the sign boards is not included. The cases when another semantic object is chosen as the target semantic object are not shown here. Following is an example showing how to use WSM to help spatial database queries.

• **Query:** *Find the video clip beginning with a player on the left of the soccer field (ground) followed by the ball appearing in the center of the ground, and then the ball disappearing and the goal line appearing on the right of the ground.*

In this query, first, we want to find a player on the left of the ground; root node with number 10 (represented *left*) of WSM is identified. The only intermediate node is checked. This intermediate node has ordered links pointing to the *Player* and *Ground* semantic object nodes. The order links pointing to *Player* and *Ground* have order number 1 and 2, respectively. This tells us that the *Player* is at the left of the target semantic object *Ground*. The corresponding frame numbers stored in this intermediate node can help us to find the query video clip that matches the first query criterion. The same mechanism is applied for the second and the third query criteria.

5. CONCLUSIONS

Traditional relational database systems consist only of alphanumeric data. The multimedia database systems not only have the alphanumeric data but also the data that cover multi-dimension spaces such as image and video data. It is very important for a database system to have an index mechanism to handle spatial data efficiently.

In this paper, a database searching structure called *WSM* that incorporates an image processing technique (*SPCPE*) is proposed to efficiently answer the multimedia database queries related to the temporal and spatial relations of the objects on the images or video frames. The Web spatial model (WSM) uses the basic twenty-seven spatial relations to model the spatial relations among semantic objects, each covered by a bounding box. WSM has non-unique parents and allows parallel searches and concurrent browsing paths. If the spatial relations are structured in WSM, then the burden of on-line processing of the raw image or video data for the database queries involving the spatial relations is reduced.

ACKNOWLEDGMENTS

This work has been partially supported by National Science Foundation under grant NSF-MII 592101500 (Shu-Ching Chen) and partially supported by National Science Foundation under contract IRI 9619812 (Srinivas Sista, Mei-Ling Shyu, and R.L. Kashyap).

REFERENCES

1. C.A. Bouman and B. Liu, "Multiple Resolution Segmentation of Textured Images, " *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 13, pp. 99-113, 1991.
2. S.K. Chang, C.W. Yan, D.C. Dimitroff, and T. Arndt, "An Intelligent Image database System," *IEEE Trans. on Software Engineering*, vol 14, No. 5, pp. 681-688, May 1988.
3. C.Y. Roger Chen, D.S. Meliksetian, Martin C-S, Chang, L. J. Liu, "Design of a Multimedia Object-Oriented DBMS," *ACM Multimedia Systems Journal*, vol. 3, pp. 217-227, November 1995.
4. Shu-Ching Chen and R. L. Kashyap, "Augmented Transition Networks as Semantic Models for Multimedia Presentations, Multimedia Database Searching, and Multimedia Browsing," Technical Report TR-ECE 98-15, School of Electrical and Computer Engineering, Purdue University, December 1998.
5. Shu-Ching Chen and R. L. Kashyap, "A Spatio-Temporal Semantic Model for Multimedia Presentations and Multimedia Database Systems," accepted for publication in *IEEE Transactions on Knowledge and Data Engineering*.
6. J.D. Courtney, "Automatic Video Indexing via Object Motion Analysis," *Pattern Recognition*, vol. 30, no. 4, pp. 607-625, 1997.
7. A.M. Ferman, B. Gunsel, and A.M. Tekalp, "Object Based Indexing of MPEG-4 Compressed Video," in *Proc. SPIE: VCIP*, pp. 953-963, vol. 3024, San Jose, USA, February 1997.
8. B.K.P.Horn and B.G. Schunck, "Determining Optical Flow," *Artificial Intelligence*, vol. 17, pp. 185-203, 1981.
9. R. L. Kashyap and S. Sista, "Unsupervised Classification and Choice of Classes: Bayesian Approach," Technical Report TR-ECE 98-12, School of Electrical and Computer Engineering, Purdue University, July 1998.
10. S. Lakshmanan and H. Derin, "Simultaneous Parameter Estimation and Segmentation of Gibbs Random Fields using Simulated Annealing, " *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 11, no. 8, pp. 799-813, 1989.
11. S. Sista and R. L. Kashyap, "Unsupervised video segmentation and object tracking," in *IEEE Int'l Conf. on Image Processing*, October 24-28, 1999.